Correntropy-based Robust Multilayer Extreme Learning Machines

Chen Liangjun, Paul Honeine, Qu Hua, Zhao Jihong, Sun Xia

Correntropy-based Robust Multilayer Extreme Learning Machines

Chen Liangjun[a], Paul Honeine[b], Qu Hua[a], Zhao Jihong[a,c], Sun Xia[d,*]

[a]*School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, P.R.China, 710049*
[b]*LITIS Lab, Université de Rouen Normandie, Saint Etienne du Rouvray, 76800, France*
[c]*School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, P.R.China, 710061*
[d]*School of Information Science and Technology, Northwest University, Xi'an, P.R.China, 710069*

Abstract

In extreme learning machines (ELM), the hidden node parameters are randomly generated and the output weights can be analytically computed. To overcome the bad feature extraction ability of the shallow architecture of ELM, the hierarchical ELM has been extensively studied as a deep architecture with multilayer neural network. However, the commonly used mean square error (MSE) criterion is very sensitive to outliers and impulsive noises, generally existing in real world data. In this paper, we investigate the correntropy to improve the robustness of the multilayer ELM and provide sparser representation. The correntropy, as a nonlinear measure of similarity, is robust to outliers and can approximate different norms (from ℓ_0 to ℓ_2). A new full correntropy based multilayer extreme learning machine (FC-MELM) algorithm is proposed to handle the classification of datasets which are corrupted by impulsive noises or outliers. The contributions of this paper are three-folds: (1) The MSE based reconstruction loss is replaced by the correntropy based loss function; In this way, the robustness of the ELM based multilayer algorithms is enhanced. (2) The traditional ℓ_1-based sparsity penalty term is also replaced by a correntropy-based sparsity penalty term, which can further improve the performance of the proposed algorithm with a sparser representation of the data. The combination of (1) and (2) provides the correntropy-based ELM autoencoder. (3) The FC-MELM is proposed by using the correntropy-based ELM autoencoder as a building block. It is notable that the FC-MELM is trained in a forward manner, which means fine-tuning procedure is not required. Thus, the FC-MELM has great advantage in learning efficiently when compared with traditional deep learning algorithms. The good property of the proposed algorithm is confirmed by the experiments on well-known benchmark datasets, including the MNIST datasets, the NYU Object Recognition Benchmark dataset, and the Moore network traffic dataset. Finally, the proposed FC-MELM algorithm is applied to address Computer Aided Cancer Diagnosis. Experiments conducted on the well-known Wisconsin Breast Cancer Data (Diagnostic) dataset are presented and show that the proposed FC-MELM outperforms state-of-the-art methods in solving computer aided cancer diagnosis problems.

Keywords:

Deep learning, extreme learning machine, correntropy, unsupervised feature learning, computer aided cancer diagnosis

1. Introduction

Extreme learning machine (ELM) [1], as a fast and effective solution of single hidden layer feed-forward networks (SLFNs), has been increasingly investigated in recent years. Because of its extremely fast training speed, good generalization ability and the proved universal approximation/classification capability, ELM has efficiently achieved excellent learning accuracy in many applied fields, such as face recognition [2, 3, 4], image classification [5, 6, 7], text categorization [8, 9], nonlinear model identification [10] and time series prediction [11]. Many variants of ELM have been proposed in recent years, such as the regularized ELM [12], the kernel-based ELM [13], the ELM based on a regularized correntropy criterion [14] and the online sequential ELM [15], to name a few. Although, the training speed of ELM is swift when compared with the traditional gradient-based learning algorithms, it may need more hidden neurons due to the random determination of the input weights and hidden biases. Thus, several researchers have been working to address and solve this issue. In [16], Zhu et al. proposed an ELM based learning algorithm named evolutionary extreme learning machine (E-ELM) that combines ELM and differential evolutionary (DE) algorithms. By making use of a modified form of DE, E-ELM optimizes the input weights and hidden biases to avoid a swollen network structure. All of these approaches are relying on ELM with its single hidden layer architecture. It is worth noting that the concept of randomness in parameters of neural networks is not new, and goes back to the beginning of the nineties, where Pao et al. [17] proposed a random vector version of the functional-link (RVFL) net. Technically, the architecture of RVFL is different from ELM as it uses a direct link between the input layer and the output layer.

Learning relevant features from original data to achieve better representations and using the representations for classification has been significantly common in modern machine learning. As the most popular algorithms in machine learning, the deep learning algorithms [19, 20, 21] take advantage of multilayer models to find the complex structure and learn the high level representations through multiple levels of abstraction. It has been proven that deep learning algorithms outperform SLFNs in many applications [22, 23]. Unfortunately, deep architectures in neural networks always spontaneously increase the training time, and often

*Corresponding author.

require appropriate hardware (e.g. GPU). For these reasons, efforts have been made to combine the ELM and the deep architectures in order to simultaneously improve the performance and reduce the training time. In [24], Kasun et al. proposed a multilayer learning architecture which uses ELM-based autoencoder to build the deep network. To further improve the classification performance of the multilayer ELM, Jiexiong et al. [25] proposed a hierarchical ELM (H-ELM) algorithm. In H-ELM, the classifier is changed into an original ELM which can keep the universal approximation capability by exploiting the advantage of random projections. Moreover, an ELM sparse autoencoder is proposed. With this new sparse building block, H-ELM can get a sparser and more compact representation of the original data and achieve a better performance. However, the ELM-based multilayer algorithms is badly influenced by different kinds of noises, especially impulsive noise and outliers, because of a major factor: the used loss function is the mean square error (MSE), which is very sensitive to these noises. Moreover, the training is often simplified when dealing combining ELM and deep architectures. Thus, the improvement of robustness of the ELM-based multilayer algorithms should be further investigated.

The correntropy, a special case of the cross entropy in information theoretic learning [26], is a nonlinear and local similarity measure which shows the similarity between two random variables in a neighborhood of the joint space controlled by the bandwidth of the used kernel function. Compared with the MSE, one of the great ability of correntropy criterion is its insusceptibility to outliers [27, 28]. Thus, the maximization of the correntropy criterion (MCC) has been increasingly used to replace the minimization of the MSE in handling the heavy-tailed impulsive noises and outliers [29]. In [30], Wang et al. proposed a local correntropy based K-means clustering to address the real-world image segmentation problem which suffers from unknown noise and intensity inhomogeneity. In [31], Chen et al. proved that the maximum correntropy estimation is equivalent to a smoothed maximum a posteriori estimation. Generally, the correntropy uses a Gaussian kernel as its kernel function, because the Gaussian kernel is smooth and strictly positive. When using the Gaussian kernel, correntropy can induce a nonlinear metric called the correntropy induced metric (CIM) which can approximate different norms, from ℓ_0 to ℓ_2. Seth et al. used in [32] the CIM as an approximation of the ℓ_0-norm to find the sparsest vector for the compressed signal reconstruction. When used as an ℓ_0 approximator, the CIM needs fewer measurements to represent the same signals. Singh et al. proposed in [33, 34] a correntropy-based cost function for training neural network classifier, denoted C_{loss}. Essentially, C_{loss} is an MSE in some reproducing kernel Hilbert space (RKHS). Two major works have considered this correntropy-based cost function function for stacked auto-encoders (SAE) in deep learning. In [35], Qi et al. used the maximum correntropy criterion (MCC) to reduce the bad influence from outliers in deep learning.

In [36], Chen et al. proposed a correntropy-based stacked auto-encoders (CSAE) which replaces the two original terms in SAE with the C_{loss}-based loss term for the fitness and the C_{loss}-based sparsity penalty term. CSAE shows an obvious improvement of the robustness when data contains outliers and impulsive noise. The combination of the two correntropy-based terms for fitness and sparsity is designated by full correntropy in the following.

Moreover, deep learning algorithms have been recently investigated in computer aided cancer detection and diagnosis (CAD) to help doctors to achieve correct examination. With the progress of deep learning algorithms in image recognition areas, convolutional neural networks (CNN) and other deep learning methods have been recently investigated for medical imagery, in particular cancer tissue images. By utilizing deep learning algorithms, some promising results on breast cancer image analysis are presented in [37, 38, 39, 40]. Similarly, CNN is used in [41, 42] to detect colon cancer by taking advantage of the good feature extraction ability in deep neural networks. Deep learning algorithms are also applied to help detecting prostate cancer and other cancers [44, 45]. Moreover, they fulfill their potential in skin cancer diagnosis with photographs, which is a difficult problem in the past. In [43], the Google's Inception v3 architecture is used for dealing with a dataset of skin photographic images, yielding excellent results.

In complement to medical imagery, some statistic data from patients are also used in CAD to aid practitioners in early detection of cancers. With the rapid development of deep learning algorithms, some great results based on deep learning methods are also presented in early CAD researches. In [46], the authors utilized the good feature extraction ability of deep-belief network (DBN) to deal with CAD for early detection of breast cancer. When compared with C4.5 decision tree method, supervised fuzzy clustering and some neural network based methods, DBN shows the best performance. Deep learning methods are also employed in gene expression analysis for cancer detection. While it is difficult to identify the subtype of a cancer, this information is of great influence in the selection of the specific treatment. In [47], Rasool et al. applied unsupervised feature learning to handle the unlabeled data to enlarge the dataset and use deep learning methods to accurately classify the cancer. Muxuan et al. proposed a DBN based multi-platform to cluster multi-platform observation data form cancer patients and to distinguish the subtypes of the cancer [48]. statistic data and genetic information from patients can provide effective information for early cancer prediction and detection. But compared with medical image based CAD, these areas still need further investigation.

In this paper, we propose a full correntropy-based multilayer extreme learning machine (FC-MELM). To this end, we replace firstly the MSE-based loss function in the conventional ELM autoencoder by the

correntropy-based loss function. Secondly, the sparsity penalty term in ELM autoencoder is also changed into a correntropy-based sparsity penalty term. These two correntropy-based terms are combined to form the full correntropy-based ELM autoencoder. Thirdly, the full correntropy-based ELM autoencoder is investigated to build the FC-MELM. These ingredients guarantee that FC-MELM is robust to impulsive noise and outliers. In order to corroborate these theoretical assumptions, the good performance of the proposed algorithm is confirmed by extensive experimental results on well-known benchmarks collected from totally different fields, including the MNIST dataset [53], the NYU Object Recognition Benchmark (NORB) dataset [54] and the Moore network traffic dataset (Moore). Then, the proposed FC-MELM is applied to classify Wisconsin Breast Cancer Data (Diagnostic) as a CAD application for early cancer detection. Hence, FC-MELM is proved to be an algorithm that can be employed in many areas to achieve relevant results. In addition, by using ELM-based building block, the FC-MELM is also faster than other traditional deep learning algorithms. Therefore, FC-MELM is suitable for solving time-sensitive problems as well.

The rest of the paper is organized as follows. In the next section, the related works are concisely introduced. And the new model FC-MELM is presented in section 3. In section 4, experiments are carried out to show the improvements of performance of the proposed FC-MELM. Section 5 provide experiment results for Computer Aided Cancer Diagnosis. Finally, section 6 concludes this paper.

2. Related Works

To help understand the FC-MELM, a brief review of the original ELM, its universal classification capability, the H-ELM and the correntropy is given in this section.

2.1. Classical ELM

Generally, ELM is a rapid solution of SLFNs by analytically determining the output layer weights. Following the presentation in [49], for a SLFN with a hidden layer of L nodes, when given N distinct samples $\{x_j, t_j\}_{j=1}^{N}$ where $x_j \in R^d$ is an input sample and $t_j \in R^c$ is the corresponding target, the output vector is defined as follows

$$f(x_j) = \sum_{i=1}^{L} \beta_i \, h(x_j, w_i, b_i), \text{ for all } j = 1, 2, ..., N, \tag{1}$$

where $h(\cdot)$ is the activation function, w_i , b_i and β_i denote the hidden layer weight, bias and output weight for the i-th unit, respectively. This is the "generalized" SLFNs with any type of hidden node $h(x_j, w_i, b_i)$. Specially, we use in this paper the additive hidden, namely

$$f(x_j) = \sum_{i=1}^{L} \beta_i \, h(w_i \cdot x_j + b_i), \text{ for all } j = 1, 2, ..., N, \tag{2}$$

where $w_i \cdot x_j$ denotes the inner product between the hidden layer weight w_i and the input x_j. In ELM, the values of the weights w_i and biases b_i are randomly determined without tuning procedure. By regrouping the tunable parameters, which are the output weights between the hidden layer with L nodes and the output layer, namely $\beta = (\beta_1, \ldots, \beta_L)^T$, the above N expressions can be written in matrix form as

$$F = H\beta, \tag{3}$$

where H is the hidden layer output matrix, with

$$H = \begin{pmatrix} h(w_1 \cdot x_1 + b_1) & \ldots & h(w_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ h(w_1 \cdot x_N + b_1) & \ldots & h(w_L \cdot x_N + b_L) \end{pmatrix}. \tag{4}$$

Thus, by minimizing the fitness error in the squared error sense, the matrix of the weights between the hidden layer and the output layer can be achieved, namely

$$\min_{\beta} ||H\beta - T||^2, \tag{5}$$

where $||\cdot||$ denotes the Frobenius norm and $T = (T_1, \ldots, T_L)^T$ is the matrix of targets. The optimal solution to (5) is given by

$$\beta^* = H^{\dagger}T, \tag{6}$$

where $H^{\dagger}$ denotes the Moore-Penrose generalized inverse, defined by the following expression when $H^T H$ is nonsingular:

$$H^{\dagger} = (H^T H)^{-1} H^T. \tag{7}$$

Obviously, a disadvantage of (5) is that the least square solution is sensitive to noise, especially to outliers and impulsive noise. It turns out that such types of noise are common in the real world datasets.

2.2. Universal Classification Capability of ELM

In this section, the sufficient and necessary conditions for universal classification of ELM are presented.

1) Universal approximation: In [50], it is proved that ELM can approximate any continuous target function by using an extensive type of feature mapping $h(x)$. For any target continuous function $f(x)$, there exists a series of β_i's such that

$$\lim_{L \to +\infty} ||f_L(x) - f(x)|| = \lim_{L \to +\infty} \left|\left| \sum_{i=1}^{L} \beta_i h_i(x) - f(x) \right|\right| = 0. \tag{8}$$

With this universal approximation capability, there is no necessity for fine-tuning. The optimization constraints of ELM are also gentler than others. These advantages will increase the generalization ability and also reduce the computational complexity. Additionally, it is known that a learning algorithm can't approximate all continuous target function by using a feature mapping which doesn't meet the universal approximation condition. Therefore, the universal approximation condition is the sufficient and necessary condition for the common used feature mappings.

2) Classification: It is shown in [51], and earlier and more generally in [52], that the classification capability of the "generalized" SLFNs with the hidden-layer mapping $h(x)$ satisfies the condition (8), by proving a theorem as follows: Given a feature mapping $h(x)$, if $h(x)\beta$ is dense in $C(R^d)$ or in $C(M)$, where M is a compact set of R^d, then a "generalized" SLFN with such a hidden-layer mapping $h(x)$ can separate arbitrary disjoint regions of any shapes in R^d or M. In the theorem, a closed set is defined as a region. It can be seen from this theorem that if the number of hidden nodes L in a classifier is sufficient, the output of this classifier $h(x)\beta$ can be close to the labels.

2.3. Variants of ELM

To improve the stability and performance of the original ELM, the regularized ELM was proposed in [12], with the solution changed to

$$\beta^* = \left(H^T H + \lambda I\right)^{-1} H^T T, \tag{9}$$

where λ is a tunable regularization parameter and I is the identity matrix of appropriate size. This formulation allows to overcome the case when $H^T H$ is singular. It is worth noting that the solution in (7) can be achieved by solving the minimization problem:

$$\min_{\beta} ||H\beta - T||^2 + \lambda||\beta||^2, \tag{10}$$

where $||\beta||^2 = \sum_{i=1}^{L} \beta_i^2$ denotes the regularization term with a ℓ_2-norm.

An autoencoder extracts features through encoding the original inputs, while approximate outputs are generated from the decoding process by minimizing the reconstruction errors. Thus, the autoencoder aims to learn a function $h_{\{W,b\}}(x) \approx x$, where W being the hidden weights and b the bias. The ELM sparse autoencoder proposed in [25] combines the ELM and a sparsity constraint. It has been proved that ELM has the universal approximation capability which helps us free from the fine-tuning procedure. Moreover, an ℓ_1-norm regularization is added to learn sparser features of the inputs. The ELM sparse autoencoder considers the optimization problem

$$O_\beta = \arg\min_{\beta} ||H\beta - X||^2 + \lambda||\beta||_1, \tag{11}$$

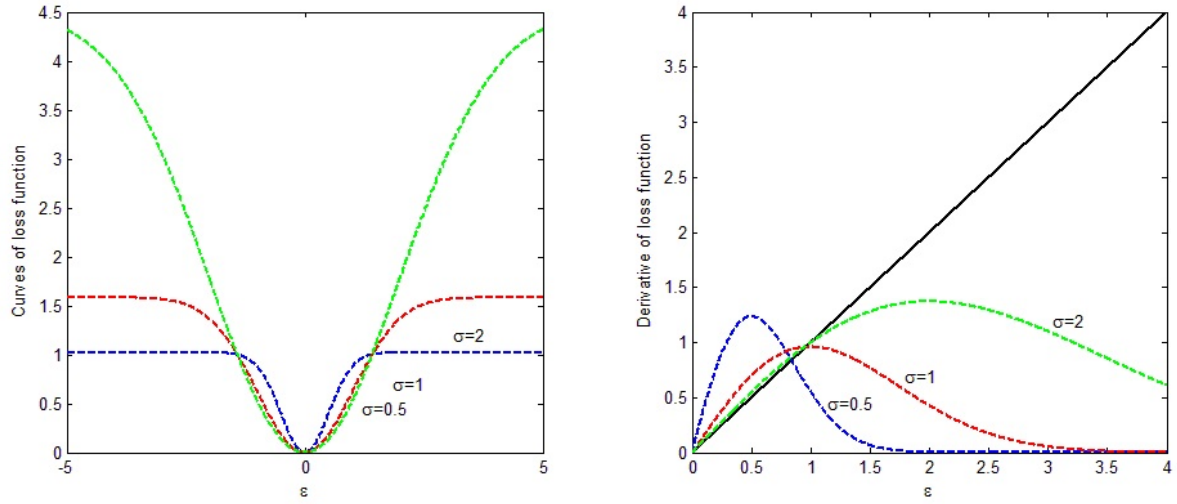

Figure 1: (a) The function value of C_{loss} for different values of the kernel bandwidth σ as the error e increasing; (b) The derivative of C_{loss} for different values of the kernel bandwidth σ, as well as the derivative of the MSE (in continuous-black).

where X is the input data. The ELM sparse autoencoder is learned by adopting a fast iterative shrinkage-thresholding algorithms (FISTA) [56]. It is noted that H is randomly initialized and does not need to be optimized.

In [25], the ELM sparse autoencoder is used as the building block to construct a hierarchical ELM (H-ELM). In the same spirit as Built in a hierarchical manner, H-ELM is different from classic deep learning architectures which are greedily layer-wisely trained. The H-ELM training process has two separate phases: 1) unsupervised hierarchical feature extraction and 2) supervised feature classification. In the first phase, the ELM sparse autoencoder is employed to extract high level features from the input data. Secondly, an original ELM-based regression is carried out for final classification. Additionally, thanks to the ℓ_1 regularization, H-ELM can further reduce the number of hidden nodes and accelerate the training process.

2.4. Correntropy

The correntropy is a generalized correlation function in information theoretic learning. It is defined for two random variables S and T as

$$C(S,T) = E[\kappa(S,T)], \tag{12}$$

where E is the expectation and $\kappa(\cdot,\cdot)$ is an arbitrary kernel function satisfying Mercer's theorem. Often used in the literature, the Gaussian kernel is defined for any two samples s_i and t_i as

$$\kappa_\sigma(s_i,t_i) = \exp\left(-\frac{e^2}{2\sigma^2}\right), \tag{13}$$

where σ is the tunable bandwidth of the kernel and $e = s_i - t_i$. Then, sample estimator of the correntropy is given by

$$\widehat{C}(S,T) = \frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(s_i,t_i). \tag{14}$$

From the perspective of functional analysis, the kernel bandwidth determines the inner product, i.e., the metric of similarity in RKHS. On one hand, if the kernel bandwidth is too large, then all the data would look similar in the RKHS (with inner products all close to 1). On the other hand, if the kernel bandwidth is too small, then all the data would look different (with inner products all close to 0). In essence, the correntropy is a second-order statistic of the mapped data into the feature space.

In [27], the probabilistic meanings of correntropy is presented as follows.

Property: Assume i.i.d. data $\{(s_i,t_i)\}_{i=1}^N$ are drawn from the joint probability density function (pdf) $f_{\mathbf{S},\mathbf{T}}(s,t)$, and $\hat{f}_{\mathbf{S},\mathbf{T};\sigma}(s,t)$ its Parzen estimate with kernel size σ. The correntropy estimate with kernel size $\sigma' = \sqrt{2}\sigma$ is the integral of $\hat{f}_{\mathbf{S},\mathbf{T};\sigma}(s,t)$ along the line $s = t$

$$\hat{V}_{\sqrt{2}\sigma}(\mathbf{S},\mathbf{T}) = \int_{-\infty}^{\infty}\hat{f}_{\mathbf{S},\mathbf{T};\sigma}(s,t)\mid_{s=t=u} du. \tag{15}$$

Then, based on the conditions of the Parzen method, as σ gets close to zero and the N_σ to infinity, $\hat{f}_{S,T;\sigma}(s,t)$ approaches the true pdf $f_{S,T}(s,t)$, we have

$$\begin{aligned}\lim_{\sigma\to 0} V(S,T) &= \lim_{\sigma\to 0}\int\int_{s,t}\kappa_\sigma(s-t)f_{S,T}(s,t)dsdt \\ &= \int\int_{s,t}\delta(s-t)f_{S,T}(s,t)dsdt \\ &= \int_{s=-\infty}^{+\infty} f_{S,T}(s,s)ds.\end{aligned} \tag{16}$$

In practice, we can only achieve an estimation of the correntropy with a finite number of samples. The kernel size is set with a lower bound, that if the value of kernel size is too small, the estimation is unmeaning at all. If σ is the kernel size of correntropy, then the bandwidth of its rectangle approximation is $\sqrt{\pi/2\sigma}$. We can also assume that the joint pdf is smooth in this bandwidth. Therefore, another interpretation of the correntropy is

$$V_\sigma(S,T) \approx P(|T-S| < \sqrt{\pi/2\sigma})/\sqrt{\pi/2\sigma}. \tag{17}$$

In classification tasks, the goal is minimizing the loss function between the classifier output variable S and the target label T. Hence, the correntropy-based loss function is defined as follows:

$$C_{\text{loss}}(S,T) = \gamma\big(1 - E(\kappa_\sigma(S,T))\big), \tag{18}$$

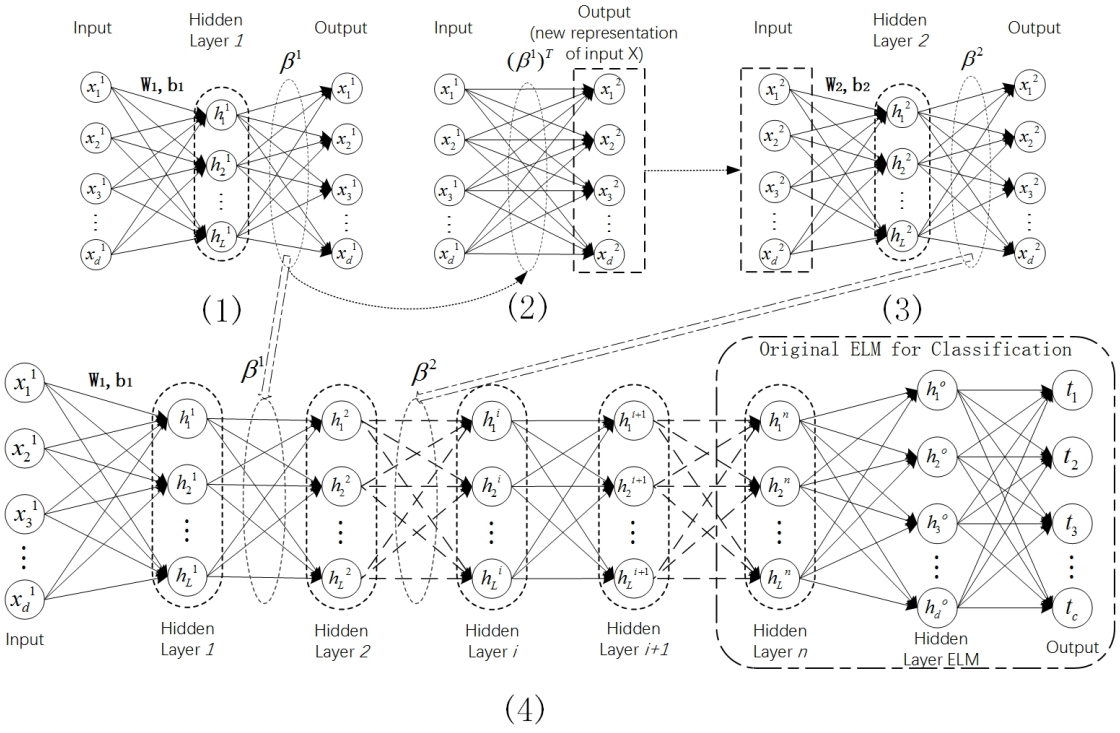

Figure 2: The architecture of FC-MELM. The full correntropy-based ELM autoencoder is used as the building block to form the FC-MELM. (1) β^1 is the output weight vector between hidden layer 1 and hidden layer 2. (2) The new input x^2 is achieved by $g(x^1 \cdot \beta^{1T})$. (3) x^2 is the input data of hidden layer 2 and used to train the second hidden layer to achieve the newer representation. (4) Once the features are learned, the output of last hidden layer is used to train an original ELM and calculate the output weight β^o.

where $\gamma = \left(1 - \exp(-\frac{1}{2\sigma^2})\right)^{-1}$. Similarly, the empirical loss function $\widehat{C}_{\text{loss}}$ can be easily achieved as

$$\widehat{C}_{\text{loss}}(S,T) = \gamma\left(1 - \frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(s_i,t_i)\right). \tag{19}$$

This loss function should be compared to the conventional MSE, namely $\frac{1}{N}\sum_{i=1}^{N} e_i^2$ with $e_i = s_i - t_i$.

In Figure 1, the curves and the derivative curves of C_{loss} are presented. In Figure 1(b), the derivative curves of MSE is also expressed to show the advantage of C_{loss}. Clearly, when the error e increases, the derivative of MSE will increase linearly. On the contrary, the derivative curves of C_{loss} decrease fast and get close to zero, which means the C_{loss} can lighten the bad influence from the outliers or impulsive noises, making it a robust alternative to MSE.

3. Description of the full correntropy-based multilayer ELM

This section describes the proposed full correntropy-based multilayer ELM (FC-MELM). To this end, it architecture is firstly presented. Secondly, the main building block is described, namely the full correntropy-based ELM autoencoder. Finally, the building of the FC-MELM is described in detail.

3.1. Architecture of FC-MELM

FC-MELM is built in a multilayer manner, in the same spirit as the H-ELM. It is composed of two parts. The first part is a multilayer feature extractor representing a process of unsupervised learning. Each layer is based on the full correntropy-based ELM autoencoder. In the second part, an original ELM operates at the output of the first part and acts as a classifier to obtain the final classification results. The target is getting a new representation through training the full correntropy-based ELM autoencoder. The deep network is fully connected with n hidden layers and $\beta = (\beta^1, \ldots, \beta^n)^T$ is the output weight to be learned from the training procedure. Each layer is trained layer-wisely to cut down the training cost. The detailed architecture and learning process are presented in Figure 2. As is shown in Figure 2(1), β^1 is learned by considering a corresponding full correntropy-based ELM autoencoder with the target output being the input, namely $T = X$. In Figure 2(2), the new input x^2 is achieved by $g(x^1 \cdot (\beta^1)^T)$. Then, in Figure 2(3), x^2 is used as input to the next hidden layer and the new output weight β^2 is learned from the new representation of input X. In Figure 2(4), the learning process of the FC-MELM from the training set X is illustrated. Identically, the β^{i+1} is the $(i+1)$-th layer output weight. At the end, an ELM-based classifier is learned from the output of n-th hidden layer.

The output matrix of each hidden layer can be denoted as

$$H_i = g\left(H_{i-1} \cdot \beta_i\right), \tag{20}$$

where H_i is the output matrix of the i-th hidden layer, β_i represents the output weights, and $g(\cdot)$ is the activation function such as the sigmoid function. The hidden layers are layer-wisely trained. Different from the traditional deep learning algorithms, when the feature extraction of the previous hidden layer is done, the weights or parameters of the current hidden layer is fixed without any fine-tuning in the FC-MELM. Thus, the FC-MELM maintains a fast training speed, as opposed to other deep architecture algorithms.

3.2. Full correntropy-based ELM autoencoder

In this section, we describe the main building block of the FC-MELM, which is the full correntropy-based ELM autoencoder.

3.2.1. Correntropy-based reconstruction loss term

In order to improve the robustness of original ELM autoencoder, we consider a correntropy-based reconstruction loss term to utilize the insensitiveness of the correntropy of the outliers and impulsive noises. Here, we maximize the correntropy between the target and network output variables instead of using cost function in (9). Therefore, we can get the correntropy reconstruction loss from the ELM sparse autoencoder as follows

$$O_\beta = \arg\min_\beta E(\kappa_\sigma(H\beta, X)) - \lambda||\beta||_1. \tag{21}$$

Compared with the existing ELM autoencoder which simply uses MSE as the reconstruction loss function, the use of the correntropy reconstruction loss for ELM autoencoder is more robust to non-Gaussian noise. By decreasing the bad influence from outliers or impulsive noises, it would have a better performance, as shown in experiments.

3.2.2. Correntropy-based sparsity penalty term

As aforementioned, C_{loss} can approximate norms from ℓ_0 to ℓ_2. For one-dimensional samples, $C_{\text{loss}}(X, \mathbf{0})$ can approximate the ℓ_0 norm as $||X||_0$, with

$$\|X\|_0 \approx C_{\text{loss}}(X, \boldsymbol{0}), \tag{22}$$

where C_{loss} is defined in (18) and empirically in (19), with $\sigma \to 0$. Generally, minimizing an ℓ_0 norm is a way to find the sparsest representation of the original data. Unfortunately, ℓ_0 norm minimization is an NP-hard problem. Instead, the ℓ_1 norm minimization is usually utilized, as with sparse autoencoders. Here, we replace the ℓ_1 norm minimization by a C_{loss}-based ℓ_0 norm approximator, which can be arbitrarily close to the ℓ_0 norm as $\sigma \to 0$. By minimizing the C_{loss}-based ℓ_0 norm approximator, one may need less measurements to represent same data when compared with ℓ_1 norm minimization, and a sparser representation can be obtained as well.

Hence, we replace in the sparse autoencoder the traditional sparsity penalty term by a correntropy-based sparsity penalty term. Then, the ℓ_1-based penalty term $||\beta||_1$ in ELM sparse autoencoder is changed to

$$||\beta||_0 \approx C_{\text{loss}}(\beta, \mathbf{0}), \tag{23}$$

where C_{loss} is defined in (18) and empirically in (19), with $\sigma \to 0$. Using correntropy-based sparsity penalty term to train the ELM autoencoder, we may get a sparser representation of the original data. Because the sparsest representation reflects the most essential features of the data which may not be sensitive to outliers, the sparser representation is helpful to increase the robustness and performance of the full correntropy-based ELM autoencoder.

Algorithm 1 Unsupervised training procedure of FC-MELM using FISTA

Input: Input matrix $X = (s_{j,i})_{N\times d}$, target matrix $T = (t_{j,k})_{N\times c}$

Output: The optimal weight matrix $\beta = (\beta_{p,q})_{L\times N_l}$

Initialization: Number of hidden layers N_l, number of hidden layer units L_n, maximum number of iterations I, kernel bandwidth of hidden layers σ_n

Randomly initialize the weight vectors connecting the input and hidden layers W,

for $\zeta = 1, 2, \ldots, I$ **do**

  Compute the Lipschitz constant φ of the gradient of smooth convex function $\bigtriangledown p(\cdot)$

  for $m = 1, 2, \ldots, I$ **do**

    Compute $\beta_m = s_\varphi(y_m)$, where

$$s_\varphi = \arg\max_{\beta} \frac{\varphi}{2}\left\|\beta - \left(\beta_{m-1} - \frac{1}{\varphi} \bigtriangledown p(\beta_{m-1})\right)\right\|^2 - q(\beta). \tag{24}$$

    Update $t_{m+1} = \frac{1+\sqrt{1+4t_m^2}}{2}$.

    Update $y_{m+1} = \beta_m + \frac{t_{m-1}}{t_{m+1}}\left(\beta_m - \beta_{m-1}\right)$.

  end for

end for

return result

3.2.3. Building the FC-MELM

To further improve the robustness, we use correntropy-based reconstruction loss and correntropy-based sparsity penalty term together to build a full correntropy-based ELM autoencoder. And this newly proposed full correntropy-based ELM autoencoder is employed as the building block to construct the FC-MELM. The cost function of the full correntropy-based ELM autoencoder is defined as follows

$$O_\beta = \arg\max_{\beta} E(\kappa(H\beta - X)) - C_{\text{loss}}(\beta, \mathbf{0}), \tag{25}$$

where $C_{\text{loss}}(\beta, \mathbf{0})$ is the correntropy-based sparsity penalty term. To solve this optimization problem, the fast iterative shrinkage-thresholding algorithms (FISTA) [56] can also be used to obtain the optimal solution. By using FISTA, one can get a minimization of a smooth convex function with a complexity of $O(1/m^2)$, where m represents the iteration times. The FISTA implementation of FC-MELM is shown in Algorithm 1. Through the Algorithm 1, the resultant bases β, as the weights of the proposed FC-MELM feature extractor, are achieved. The learned features of the original data is saved in β. By using β, the inner product of the inputs and the learned features would reflect the compact representations of the original data. By building FC-MELM with the full correntropy-based ELM autoencoder, sparser high level feature representations can

be achieved by layer-wise comparison.

Moreover, in [32], it is proved that the C_{loss}-based ℓ_0 norm approximator has a better performance in data recovery and other applications. Its good ability is reflected on reducing the number of measurements (neural nodes). Therefore, the full correntropy-based ELM autoencoder can further cut down the computational complexity of the FC-MELM.

4. Experimental Results

This section presents the performance of the proposed FC-MELM and state-of-the-art methods. To this end the classification accuracies are studied over well-known benchmark datasets, such as handwritten datasets MNIST and USPS, NYU Object Recognition Benchmark dataset (NORB), Moore network traffic dataset (Moore). In all the experimentations below, the hardware and software conditions are listed as follows: a desktop computer equipped with an Intel-i7 6700 4.0G CPU, 16G DDR4 RAM, running MATLAB R2016b on Microsoft Windows 10.

4.1. Datasets

In order to reflect the convictive robustness of the proposed FC-MLEM, a mix-Gaussian noise is deployed in MNIST and Moore datasets as an observation noise to corrupt the original training samples. The mix-Gaussian noise randomly generates outliers with abrupt large values and thus models strong impulsive noises. It is generated as follows:

$$G(x) = (1 - A)\, g(0, v_1^2) + A\, g(0, v_2^2), \tag{26}$$

where $g(0, v_i^2)$ denotes the zero-mean Gaussian distribution with variance v_i^2 and A is the mixture coefficient that controls the noise level.

In order to more generally demonstrate the better robustness of proposed FC-MELM, we also introduce another non-Gaussian noise which is the α-stable noise. As the α-stable distribution has no closed-form expression for the probability density function, its characteristic function is used to describe it:

$$\varphi(x) = \exp\left\{j\mu x - \gamma|x|^{\alpha}\left[1 + j\beta sign(x)\, w(x, \alpha)\right]\right\}, \tag{27}$$

where

$$w(x, \alpha) = \begin{cases} \tan\left(\frac{\pi\alpha}{2}\right) & \text{for } \alpha \neq 1 \\ \frac{2}{\pi}\log|x| & \text{for } \alpha = 1 \end{cases} \tag{28}$$

where $-\infty < \mu < +\infty$ is the location parameter; $\gamma > 0$ is the dispersion parameter, which relates to the distributions spread around the center; $-1 < \beta < 1$ is the symmetry parameter, when $\beta = 0$, the distribution is symmetric about μ and it is called a symmetry α-stable distribution ($S\alpha S$); $0 < \alpha \leq 2$ is the characteristic exponent parameter, which determines the heaviness of the tails. In this paper, we assume $\alpha = 1.2$, $\beta = 0$ and $\mu = 0$, then the characteristic function boils down to

$$\varphi(x) = e^{-\gamma|x|^{\alpha}}. \tag{29}$$

The parameter γ controls the level of α-Stable noise. With the value of γ increasing, the noise becomes more severe.

Several binary and multi-class benchmarks are investigated in this paper to analyse the performance of the FC-MELM and compare it to the state of the art. The considered benchmark datasets have been commonly used in the literature. In the following, we give a succinct presentation.

The benchmark dataset MNIST is one of the most famous handwritten benchmark datasets. It contains 60,000 samples from the training set and 10,000 samples from the test set. The dataset is classified into ten classes of handwritten images, from digit 0 to digit 9. The size of the gray scaled images is 28x28 and is normalized to [0, 1]. A mix-Gaussian noise for various levels is added into training set of the MNIST dataset in order to study the robustness of the investigated methods.

The NYU Object Recognition Benchmark (NORB) dataset is also an image dataset, but much more complicated than the MNIST. It includes 50 different 3D toy objects and these object images are classified into five generic classes. The viewpoints and lighting conditions of the images are also different. In the conducted experiment, we use the same settings as Huang et al. had used in [25], as follows. In training, there are 24,300 stereo images with 25 objects. Similarly, there are 2,430 stereo images in the test set. Specially, test set has the remaining images of 25 objects.

In order to study the performance on large and complex datasets, we consider the Moore dataset from the University of Cambridge. This network traffic dataset contains 377,526 net flow samples which are classified into 12 classes, where each sample has 249 features. The number of samples in each class is shown in Table 1. Obviously, the numbers of different classes are not averaging at all. Unfortunately, this will lead to an error result that even if only the class of "WWW" is properly learned and classified, the accuracy is still very high. To avoid this issue, we extract some samples to build a relative average dataset and name Moore-a. There are 1500 samples for each class in training set of Moore-a. Particularly, the quantity of samples in three classes (multimedia, interactive and games) is too little to separately build a "class". Thus, they will be treated as zero-day traffic flows (they are "unknown" classes and would be mislabeled to a

Table 1: Statistic Information of Moore dataset.

Network Flow Class	Applications	Number of Samples	Percentage
GAMES	Microsoft Direct Play	8	0.0021%
P2P	eDonkey, BitTorrent	2094	0.5547%
MAIL	IMAP, POP,SMTP	28567	7.5669%
WWW	Web browsers	328092	86.9058%
INTERACTIVE	SSH,TELNET	110	0.0291%
MULTIMEDIA	Windows Media Player,iTunes	576	0.1526%
FTP-DATA	FTP, wget	5797	1.5355%
SERVICES	X11, DNS, IDENT, LDAP	2099	0.5560%
ATTACK	Port scans,worms,viruses	1793	0.4750%
FTP-PASV	Skype	2688	0.7120%
DATABASE	MySQL, dbase,Oracle	2648	0.7014%
FTP-CONTROL	MSN Messenger	3054	0.8090%

"known" class) and used as a kind of noise in subsequent experiments of the classification of network traffic data corrupted by mislabeled samples. Then, there are 13500 samples for training and 9000 samples for test in total. The number of different classes in Moore-a is shown in Table 2.

4.2. Algorithm Settings

All the parameters of related methods are randomly initialized and trained layer-wisely (i.e., the hidden layers are trained one by one and output from the previous layer is used as the input of the current layer). And all the experiment results are obtained by averaging the values over 50 trials to avoid uncertainty of the random initialization and the noise generation. In addition, sigmoid activation function is employed in all the algorithms.

In simulations over classic benchmark datasets, MNIST and NORB, the experimental network structures are set based on the results in [25]. Each dataset corresponds with different network architecture. The FC-MELM and H-ELM have the same network architecture. The ML-ELM has the same network architecture in hidden layers with the former two algorithms, but its classifier is different. For easy comparison, the number of hidden nodes is kept the same in ELM-based algorithms. The selection procedure of the kernel bandwidth for FC-MELM is presented later. Moreover, the structures and parameters of other learning algorithms are set experimentally.

In simulations over Moore-a dataset, the structures and parameters are also experimentally set. The

Table 2: Statistic Information of Moore-a dataset

Network Flow Class	Training Data	Test Data
GAMES	0	0
P2P	1500	594
MAIL	1500	1500
WWW	1500	1500
INTERACTIVE	0	0
MULTIMEDIA	0	0
FTP-DATA	1500	624
SERVICES	1500	599
ATTACK	1500	293
FTP-PASV	1500	1188
DATABASE	1500	1148
FTP-CONTROL	1500	1554

hidden layers of ML-ELM, H-ELM and FC-MELM have the same number of layers and nodes. Specifically, the structure of FC-MELM is set to 700-700-12000 for the MNIST, 3000-3000-15000 for the NORB, and 248-100-100-3000 for the Moore-a dataset.

4.3. Performance Analysis over Classic Benchmark Datasets

To evaluate the performance of FC-MELM, nine classification benchmark datasets are selected to carry out the experiments. All the experiments are carried out with a ten-fold cross-validation. The classification performance of FC-MELM and other methods are compared in Table 3. Obviously, an improvement has been made in the FC-MELM in comparison with the H-ELM and other methods in most of the benchmark datasets. The results prove that FC-MELM is a more robust algorithm that achieves better performance.

4.4. Performance of FC-MELM over the MNIST Dataset

4.4.1. Classification Results of MNIST Dataset

In this section, the classification accuracies of FC-MELM, H-ELM and ML-ELM over the MNIST dataset under different levels of mix-Gaussian noises are compared.

Particularly, in order to better evaluate the relevance of proposed new terms, the MSE-based reconstruction loss and the original sparsity penalty term in H-ELM are separately replaced by the correntropy-based reconstruction loss term and correntropy-based sparsity penalty term. Then, comparison of the classification

Table 3: Classification Accuracies over Classic Benchmark Datasets

Dataset	SAE	CSAE	ML-ELM	H-ELM	FC-MELM	DNN	SVM	ELM	RBF
USPS	91.13 ± 4.47	91.10 ± 3.95	95.94 ± 0.01	96.35 ± 0.17	96.41 ± 0.13	91.73 ± 1.64	94.20 ± 0.64	93.21 ± 0.62	95.67 ± 0.21
Wine	90.28 ± 2.15	91.16 ± 3.92	98.11 ± 2.00	100 ± 0.0	100 ± 0.0	96.42 ± 4.20	95.01 ± 4.40	96.11 ± 2.58	100 ± 0.0
Iris	93.02 ± 5.67	93.67 ± 4.33	94.67 ± 4.22	100 ± 0.0	100 ± 0.0	95.67 ± 7.01	98.70 ± 1.60	93.61 ± 3.94	99.10 ± 1.3
Diabetes	75.1 ± 5.47	75.23 ± 3.99	69.92 ± 8.22	79.54 ± 4.25	82.23 ± 5.43	73.38 ± 4.13	73.16 ± 5.99	68.92 ± 4.03	82.03 ± 9.33
Liver	50.29 ± 6.83	57.68 ± 4.53	71.30 ± 8.22	71.16 ± 4.39	72.46 ± 4.49	66.67 ± 6.4	71.05 ± 4.60	70.51 ± 5.46	81.76 ± 6.33
DNA	95.35 ± 6.90	96.26 ± 4.36	93.50 ± 2.53	95.89 ± 2.33	96.14 ± 1.34	94.28 ± 1.50	95.07 ± 0.70	92.18 ± 1.96	95.85 ± 0.32
Australian	66.81 ± 4.01	68.84 ± 3.93	80.58 ± 4.93	84.66 ± 3.68	85.01 ± 3.73	66.12 ± 5.36	82.13 ± 3.80	82.34 ± 4.52	82.02 ± 5.60
Letter	80.13 ± 1.29	84.64 ± 1.76	87.84 ± 0.73	90.45 ± 0.68	92.87 ± 0.70	86.84 ± 0.98	92.04 ± 0.38	87.62 ± 0.68	85.88 ± 3.38
Vowel	91.30 ± 3.87	94.40 ± 2.94	92.42 ± 3.08	92.56 ± 2.40	92.98 ± 2.56	92.59 ± 2.89	87.43 ± 3.69	92.38 ± 3.32	95.78 ± 3.42

accuracies of the H-ELM with the correntropy reconstruction loss term and the H-ELM with the correntropy sparsity penalty term, under mix-Gaussian noises are presented.

In Table 4, the classification accuracy results of FC-MELM and other methods are shown, for different levels of the mix-Gaussian noise. It shows that when the data is clean, H-ELM can achieve the highest classification accuracy. But the classification accuracies gap between FC-MELM and H-ELM are lower than 0.1 %. When compared with H-ELM, under different levels of noise, the two correntropy-based methods show obviously higher classification accuracies. After combined the correntropy-based reconstruction loss and the correntropy-based sparsity penalty term, the proposed FC-MELM shows the highest classification accuracy, no matter what degree of mix-Gaussian noise is added. The classification accuracies of FC-MELM are almost 45% higher than H-ELM at most. Thus, the FC-MELM shows a more robust performance than H-ELM, when the noise becomes severer. Additionally, the ML-ELM has a better robustness than H-ELM. Meanwhile, the classification accuracies of FC-MELM are higher than ML-ELM all the time, and FC-MELM exceeds ML-ELM 6.34% at most. The classification accuracy curves of these algorithms are

Table 4: Classification accuracies over the MNIST dataset under different noise levels

	FC-MELM	H-ELM with the correntropy reconstruction loss	H-ELM with the correntropy sparsity penalty term	H-ELM	ML-ELM
A=0	98.89 ± 0.0004	98.73 ± 0.0003	98.65 ± 0.0004	98.98 ± 0.0004	98.43 ± 0.0003
A=0.01	98.09 ± 0.0003	97.95 ± 0.0006	97.99 ± 0.0005	97.94 ± 0.0008	97.89 ± 0.0005
A=0.02	94.91 ± 0.26	94.41 ± 0.30	93.35 ± 0.25	79.30 ± 0.90	92.13 ± 0.001
A=0.03	94.49 ± 0.33	92.84 ± 0.39	87.30 ± 0.44	61.19 ± 1.39	88.60 ± 0.002
A=0.04	93.28 ± 0.67	90.04 ± 0.75	88.98 ± 0.56	51.12 ± 1.11	86.98 ± 0.004
A=0.05	92.63 ± 0.71	88.65 ± 0.78	88.10 ± 0.86	47.91 ± 1.17	86.29 ± 0.002

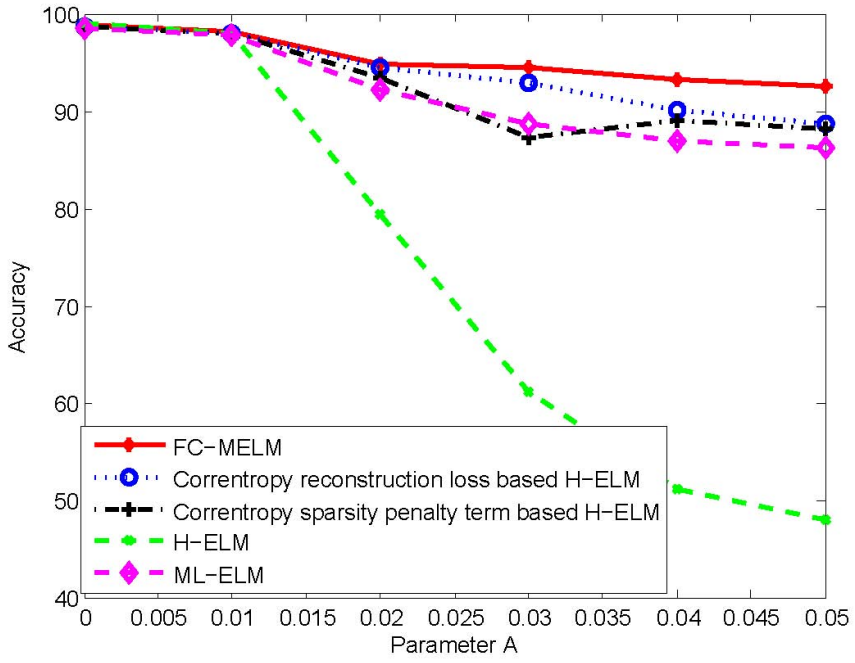

Figure 3: Comparison of classification accuracies of FC-MELM and other methods under different levels of the mix-Gaussian noise as given in (26). The accuracies of original H-ELM and ML-ELM decrease more rapidly than FC-MELM.

shown in Figure 3. Clearly, the curve of FC-MELM has the slowest decrease as the degree of noise increases. Therefore, the FC-MELM has a better robustness to the mix-Gaussian noise.

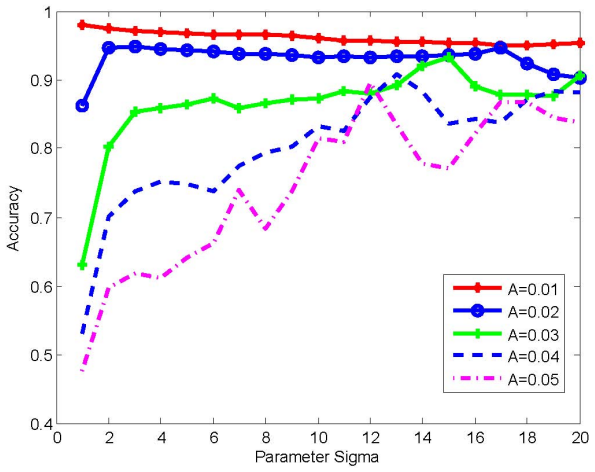

Figure 4: Influence of the bandwidth parameter $\sigma_{(1)}^{(L1)}$ of the correntropy reconstruction loss based H-ELM

4.4.2. Influence of the Bandwidth Parameter σ

The bandwidth parameter σ not only controls the shape of C_{loss} function to approximate the different norms (from ℓ_0 to ℓ_2), but also influences the robustness to outliers. Therefore, the selection of the bandwidth parameter σ is very important for both correntropy-based reconstruction loss term and the correntropy-based sparsity penalty term.

Firstly, we demonstrate the influence of the bandwidth parameter $\sigma_{(1)}^{(L1)}$ in the correntropy reconstruction loss based H-ELM. The performance of the H-ELM with the correntropy reconstruction loss depends on the value of $\sigma_{(1)}^{(L1)}$. The accuracies of classification with different values of the bandwidth parameter are shown in Figure 4, for several noise levels. While the classification accuracy can be low for important noise and small values of $\sigma_{(1)}^{(L1)}$, it tends to become high and stable when $\sigma_{(1)}^{(L1)}$ increases. It is easy to explain that when $\sigma_{(1)}^{(L1)}$ becomes too large, the C_{loss} will be approximately equivalent to MSE. H-ELM with the correntropy reconstruction loss gets the highest accuracy with different values of $\sigma_{(1)}^{(L1)}$ under different levels of noises.

Secondly, the bandwidth parameter (denoted $\sigma_{(2)}^{(L1)}$ in the following) of the correntropy-based sparsity penalty term is also examined and appropriately selected. The correntropy-based sparsity penalty term can obviously improve the robustness of H-ELM, but the effectiveness of the correntropy-based sparsity penalty term can be affected by the value of its bandwidth parameter too. Different from the bandwidth parameter selection in the correntropy reconstruction loss based H-ELM, the bandwidth parameter $\sigma_{(2)}^{(L1)}$ of the correntropy-based sparsity penalty term should be assigned a much smaller value, as explained in details in Section 2. The performance with different values of $\sigma_{(2)}^{(L1)}$ are shown in Figure 5.

Finally, the appropriately selected values of σ and σ_2 are used in the FC-MELM to accomplish the

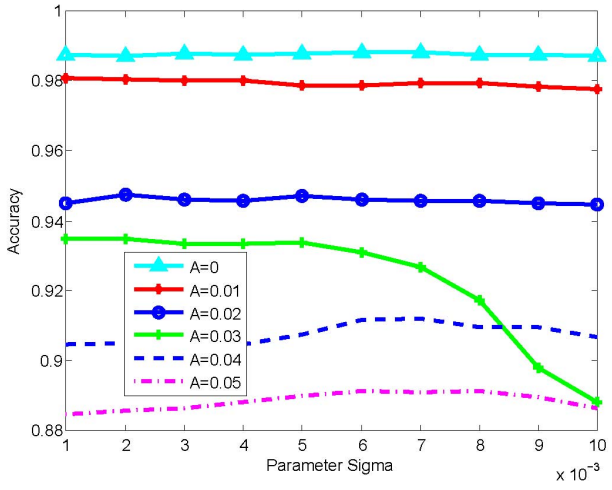

Figure 5: The influence of the bandwidth parameter $\sigma_{(2)}^{(L1)}$ of the correntropy sparsity penalty term based H-ELM, for several noise levels

Table 5: Bandwidth parameters of FC-MELM over the MNIST dataset under different noise levels

	$\sigma_{(1)}^{(L1)}$	$\sigma_{(2)}^{(L1)}$	$\sigma_{(1)}^{(L2)}$	$\sigma_{(2)}^{(L2)}$
A=0	0.9	0.006	1.2	0.003
A=0.01	1.4	0.004	1.1	0.004
A=0.02	16.8	0.002	3.6	0.006
A=0.03	15.1	0.002	3.3	0.003
A=0.04	12.7	0.007	2.4	0.006
A=0.05	11.9	0.008	2.9	0.010

classification experiments.

In Table 5, the selected values of $\sigma_{(1)}^{(L1)}$ and $\sigma_{(2)}^{(L1)}$ for the first hidden layer and second hidden layer are demonstrated. $\sigma_{(1)}^{(L1)}$ is the bandwidth parameter of the correntropy reconstruction loss term in hidden layer 1. $\sigma_{(2)}^{(L1)}$ is the bandwidth parameters of correntropy sparsity penalty term in hidden layer 1. We can see that in different hidden layers, the most suitable bandwidth parameters are different. Thus, one should carefully choose the appropriate bandwidth parameter values for each hidden layer to achieve the best results.

4.5. Performance of FC-MELM over the NORB Dataset

In this section, the NORB dataset, which is much more complicated than MNIST, is investigated in order to further study the performance of proposed FC-MELM. Particularly, we use the α-stable noise to corrupt the original dataset of NORB. The classification results of ML-ELM, the original H-ELM and the H-ELM

Table 6: Classification accuracies over the NORB dataset under different noise levels

	FC-MELM	H-ELM with the correntropy reconstruction loss	H-ELM with the correntropy sparsity penalty term	H-ELM	ML-ELM
Original dataset	91.87 ± 0.13	89.20 ± 0.25	88.79 ± 0.19	88.34 ± 0.31	88.97 ± 0.07
γ=0.001	89.09 ± 0.58	88.37 ± 0.81	87.51 ± 0.64	84.52 ± 0.73	88.05 ± 0.31
γ=0.002	88.05 ± 1.16	86.98 ± 1.46	85.88 ± 0.97	79.62 ± 0.90	86.88 ± 0.68
γ=0.003	87.10 ± 0.97	84.31 ± 1.27	82.63 ± 0.84	74.89 ± 1.39	83.79 ± 1.01
γ=0.004	86.02 ± 0.89	80.69 ± 0.99	79.49 ± 1.20	68.33 ± 1.11	80.11 ± 0.96
γ=0.005	84.87 ± 1.03	77.17 ± 1.14	76.13 ± 1.22	62.72 ± 1.17	76.43 ± 0.93

with two correntropy based terms under different levels of α-stable noise are carried out for comparison.

Table 6 shows the classification results. Obviously, classification accuracies of H-ELM deteriorates fast as the noise level increases. From these results, ML-ELM is more robust than H-ELM. The H-ELM with the correntropy reconstruction loss and the H-ELM with correntropy based sparsity penalty term achieve similar results. Finally, the proposed FC-MELM achieves the highest accuracies again under all levels of noises. Moreover, FC-MELM still obtains the highest classification accuracy on the original NORB dataset. In Figure 6, the accuracy of FC-MELM and other algorithms are shown for different noise levels. Clearly, the proposed FC-MELM maintains the highest accuracies when the noise level increases, since its curve has the slowest decrease. These experiments illustrate that FC-MELM can commendably classify the NORB dataset.

Finally, the selected bandwidth parameters are given in Table 7. One can achieve the better performance with selected bandwidth parameters.

4.6. Performance of FC-MELM over the Moore-a Network Traffic Dataset

In this section, a series simulation results are presented to demonstrate the classification ability of the FC-MELM in network traffic classification problems. As the Internet development is accelerating, a better network traffic classification is always needed to well manage the network. In the following experiments, the FC-MELM is used to classify the Moore-a network traffic dataset. Particularly, two kinds of corruption in the dataset are investigated, the mix-Gaussian noise and the mislabeled samples. For a comparative analysis, the FC-MELM is compared with H-ELM and ML-ELM to study the robustness in the network

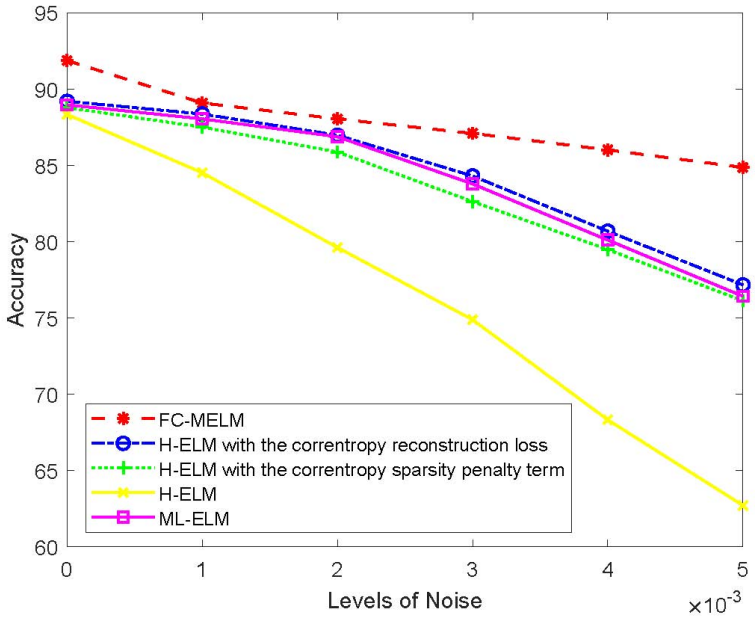

Figure 6: Comparison of classification accuracies of FC-MELM and other methods under different levels of α-stable noise as given in (27). The accuracies of original H-ELM and ML-ELM decrease more rapidly than FC-MELM.

traffic classification problem.

4.6.1. Mix-Gaussian Noise Corrupted Dataset

In this part, the classification results of FC-MELM over mix-Gaussian noise corrupted Moore-a dataset are shown. It should be pointed out that the realistic network traffic data is gathered and featured from massive original network flow data. Thus, errors may occur in every step. Particularly, when some network attacks are not recognized and cut off, they will lead to severe changes of the values of some network parameters. Thus, we use mix-Gaussian noise to simulate this kind of noise.

The performances of the FC-MELM and other algorithms are compared in Table 8. Obviously, under different noise levels, the classification accuracies of FC-MELM are always the highest. The ML-ELM is more robust than the H-ELM and the H-ELM with the correntropy reconstruction loss. When the correntropy-based sparsity penalty term is utilized, the FC-MELM still outperforms the the ML-ELM, 15.1% at most. Moreover, the FC-MELM has the highest accuracy with the uncorrupted data, which means that the FC-MELM is effective for network traffic classification as well. In Figure 7, the accuracy curves over Moore-a dataset under different levels of mix-Gaussian noises is illustrated. The curve of FC-MELM has the slowest decrease when the noise becomes severer. Therefore, it is distinct that the robust network traffic classification

Table 7: Bandwidth parameters of FC-MELM over the NORB dataset under different noise levels

	$\sigma_{(1)}^{(L1)}$	$\sigma_{(2)}^{(L1)}$	$\sigma_{(1)}^{(L2)}$	$\sigma_{(2)}^{(L2)}$
Original dataset	3.1	0.09	2.6	0.08
γ=0.001	2.3	0.06	1.8	0.04
γ=0.002	2.6	0.04	1.3	0.05
γ=0.003	1.3	0.07	2.9	0.03
γ=0.004	1.3	0.02	0.9	0.02
γ=0.005	1.1	0.01	0.7	0.03

Table 8: Classification accuracies over the Moore-a dataset under different noise levels

	FC-MELM	H-ELM with the correntropy reconstruction loss	H-ELM	ML-ELM
A = 0	94.68 ± 0.0021	94.16 ± 0.0043	92.96 ± 0.0023	92.17 ± 0.0057
A = 0.05	81.66 ± 0.0077	66.62 ± 0.0126	65.26 ± 0.0067	70.45 ± 0.0110
A = 0.1	73.44 ± 0.0048	52.19 ± 0.0101	46.92 ± 0.0098	58.21 ± 0.0059
A = 0.15	71.76 ± 0.0064	49.38 ± 0.0124	47.74 ± 0.0109	57.14 ± 0.0061
A = 0.2	71.17 ± 0.0093	47.58 ± 0.0147	42.52 ± 0.0155	56.07 ± 0.0082

can be obtained by applying the proposed FC-MELM algorithm.

In Table 9, the selected bandwidth parameters are illustrated as well.

4.6.2. Mislabeled Samples Corrupted Dataset

Computer network develops swiftly in these days and the number of applications is explosively growing. Statistically, one third of the total network traffic are generated by the new applications and named zero-day traffic. Hence, the zero-day traffic cannot be ignored any more and needs precise recognition and classification. However, the traditional network traffic classification methods are neither flexible nor robust. In such traditional methods, the classes are already settled before training. As a result, the zero-day traffic will be inevitably classified into wrong classes. These misclassified samples will in turn decrease the classification accuracies of the known classes by mis-tuning the parameters in training procedure. Thus, improving the robustness of the mislabeled samples is very crucial to the network traffic classification methods. However, although the zero-day traffic is totally unknown for the classifier, it may be partially similar to the known

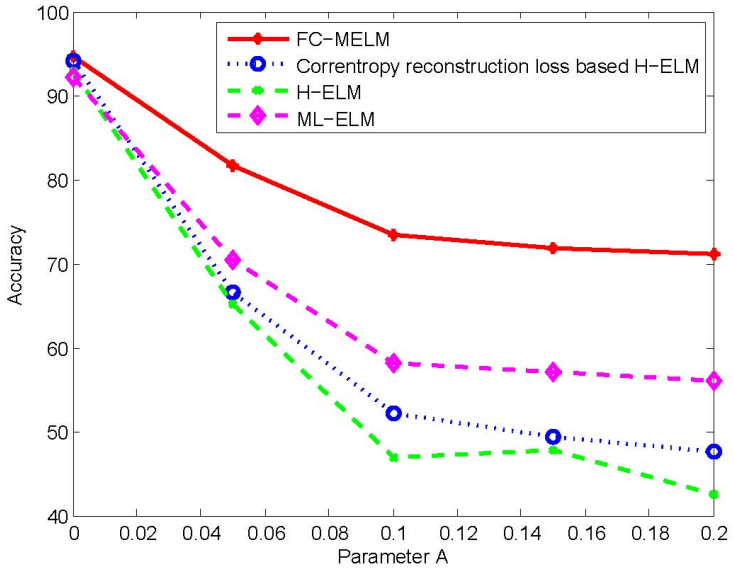

Figure 7: Comparison of classification accuracies of FC-MELM and other methods over Moore-a dataset under different levels of the mix-Gaussian noise.

Table 9: Bandwidth parameters of FC-MELM over the Moore-a dataset under different noise levels

	$\sigma_{(1)}^{(L1)}$	$\sigma_{(2)}^{(L1)}$	$\sigma_{(1)}^{(L2)}$	$\sigma_{(2)}^{(L2)}$
A=0	3	0.005	1.5	0.006
A=0.05	14.7	0.072	2.3	0.083
A=0.1	8.3	0.098	2.9	0.075
A=0.15	15.0	0.090	3.5	0.096
A=0.2	8.7	0.100	2.9	0.089

traffic. Then, we may treat the different parts between zero-day traffic and known traffic as the outliers in known traffic. Naturally, the proposed FC-MELM can be applied to deal with this classification issue by reducing the bad influence from the zero-day traffic.

In the following, we consider the Moore-a dataset corrupted with mislabeled samples. To this end, the samples from the classes "games", "multimedia" and "interactive" are considered as mislabeled. Table 10 shows the experimental results of the FC-MELM and other algorithms under different number of mislabeled samples. We can find that the H-ELM and the ML-ELM have similar accuracies. The FC-MELM always achieves the highest accuracies and outperforms H-ELM and ML-ELM, 22.36% at most. Figure 8 illustrates the accuracy curves of FC-MELM, ML-ELM and H-ELM with different number of mislabeled samples. Obviously, the accuracy curve of FC-MELM is always higher than the other methods. These experiments

Table 10: Classification accuracies over the Moore-a dataset with mislabeled samples

Number of mislabeled samples	FC-MELM	H-ELM with the correntropy reconstruction loss	H-ELM	ML-ELM
0	94.68 ± 0.0026	94.16 ± 0.0034	92.96 ± 0.0034	92.17 ± 0.0057
100	94.46 ± 0.0046	93.99 ± 0.0061	92.34 ± 0.0023	92.03 ± 0.0047
500	93.33 ± 0.0083	90.51 ± 0.0103	82.93 ± 0.0097	79.84 ± 0.0248
900	90.16 ± 0.0098	86.02 ± 0.0111	63.84 ± 0.0135	66.75 ± 0.0129

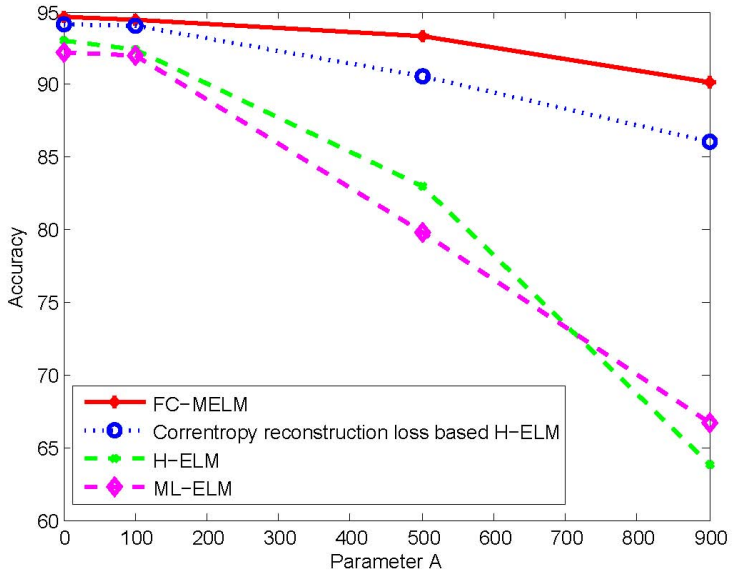

Figure 8: Comparison of classification accuracies of FC-MELM and other methods over Moore-a dataset with mislabeled samples, for different numbers of mislabeled samples.

clearly demonstrate that FC-MELM is more effective in solving classification problems of the network traffic data, includes mislabeled samples.

Finally, the selected bandwidth parameters of FC-MELM over the Moore-a dataset with mislabeled samples are still presented Table 11.

4.7. Training Time Comparison of FC-MELM and H-ELM

ELM based multi-layer learning algorithms inherits the fast training speed as well. In [25], Huang et al. proved that the training time of H-ELM is much less than other traditional deep learning algorithms. The training time of H-ELM is only one tenth to one hundredth of the traditional deep learning algorithms.

Table 11: Bandwidth parameters of FC-MELM over the Moore-a dataset with mislabeled samples

	$\sigma_{(1)}^{(L1)}$	$\sigma_{(2)}^{(L1)}$	$\sigma_{(1)}^{(L2)}$	$\sigma_{(2)}^{(L2)}$
100	1.7	0.018	0.6	0.007
500	2.5	0.024	0.5	0.009
900	2.3	0.019	0.8	0.010

Table 12: Training Time Comparison of FC-MELM and H-ELM

Datasets	MNIST	NORB	Moore-a
FC-MELM	268s	498s	82s
H-ELM	91s	131s	28s

There is a tendency in deep learning area that the network structure becomes more complicated. Thus, it is very valuable for the multi-layer ELM to achieve fast training speed. In this paper, the H-ELM is extended with correntropy. Although it is proved that correntropy based methods have much better robustness of non-Gaussian noises, these methods have higher computation complexity than the MSE based methods. In order to verify the training speed of proposed FC-MELM, the training time of FC-MELM and H-ELM in MNIST, NORB and Moore-a datasets are shown in Table 12. The training time is achieved by averaging the time over ten trials. We only present the training time of the original datasets, because the training time of the contaminated datasets is similar with that of original datasets. It is obvious that the training time of FC-MELM is a few times longer than that of H-ELM. However, when compared with the traditional deep learning algorithms, FC-MELM still has huge advantage on training time. Therefore, FC-MELM is an efficient multi-layer learning algorithm as well.

5. Application in Computer Aided Cancer Diagnosis

Early diagnosis of cancer is the prerequisite to cure cancer and the best way to save lives. But too many people missed the optimal cure time and their disease becomes chronic because of lacking reliable and propagable technology in early diagnosis of cancer. In this section, we study the breast cancer diagnostic problem. Recently, computer aided detection and diagnosis (CAD) systems are popular and achieve great progress with the advent of deep learning. However, the requirements for real-time and high accuracy cannot be satisfied at the same time so far. For instance, the recently proposed deep-belief network (DBN) in [46]

for early detection of breast cancer is so complicated that one cannot flexibly adjust or retrain a DBN based diagnosis system over a short span of time. Naturally, the factors related with occurrence of breast cancer or other cancers are changing. With the studies advancing, some factors may be added to or deleted from the dataset. Therefore, a fast training CAD method is required for large-scale data-mining of early detection of cancers. Thus, by taking advantage of rapid training speed of proposed FC-MELM, we can build a flexible CAD system which can be re-modulated or retrained in no time.

Meanwhile, in practice, cancer statistic data have a lot of sources. Different data may be collected in different formats and methods. Therefore, the lack of standardization in data collecting and preprocessing may cause many problems, such as outliers. These ubiquitous outliers may badly influence the diagnosis results. Thus, with the development of CAD, it is necessary to keep improving the robustness of the classifiers which are used in early cancer detection systems. FC-MELM has the ability of reducing the vicious influence from outliers and maybe helpful to achieve better results.

In this section, we investigate the proposed FC-MELM to solve CAD problem over WBCD. The WBCD dataset is a world famous breast cancer dataset. It is obtained from the University of Wisconsin Hospital, Madison from Dr. William H. Wolberg and has been extensively applied to test the classification ability of new proposed methods in cancer diagnostic. The dataset contains 569 instances with 30 attributes. There are 357 instances for benign and 212 instances for malignant. Although this dataset is not too large, it has sufficient representativeness to show the performance of classification algorithms in solving analogous cancer diagnosis problems. To demonstrate the robustness of the studied methods, a mix-Gaussian noise is added in the training set. The hidden layers of FC-MELM and other ELM based multi-layer models are all set to 30-100-100-3000 for the WBCD dataset.

Table 13 shows the classification accuracies of FC-MELM, H-ELM with the correntropy reconstruction loss, H-ELM and ML-ELM, over WBCD under different levels of mix-Gaussian noises. It is clear that H-ELM and ML-ELM are very sensitive to impulsive noise. The H-ELM with the correntropy reconstruction loss can get much better accuracies. It goes without saying that FC-MELM always achieves the highest accuracies no matter how the noise increases. Additionally, FC-MELM also gains a higher accuracy in classifying the original dataset, without any added noise. In Figure 9, the accuracy of FC-MELM and other algorithms are shown for different noise levels. It is easy to see that the proposed FC-MELM is the least affected when the noise level increases, since its curve has the slowest decrease. These experiments demonstrate that FC-MELM is suitable to deal with cancer computer-aided diagnosis. Thus, FC-MELM can undoubtedly be used to handle this problem well and fast. In Table 14, the training time comparison of

Table 13: Classification accuracies over the WBCD under different noise levels

	FC-MELM	H-ELM with the correntropy reconstruction loss	H-ELM	ML-ELM
A=0	96.46 ± 0.158	94.69 ± 0.103	92.04 ± 0.085	87.91 ± 0.0247
A=0.01	94.76 ± 0.328	92.92 ± 0.513	68.14 ± 0.757	60.18 ± 0.220
A=0.02	93.81 ± 0.339	88.50 ± 0.237	54.87 ± 0.631	53.25 ± 0.347
A=0.03	92.04 ± 0.328	84.07 ± 0.496	52.61 ± 0.323	51.39 ± 0.395
A=0.04	91.28 ± 0.435	83.19 ± 0.349	56.64 ± 0.450	52.21 ± 0.681
A=0.05	90.27 ± 0.457	81.42 ± 0.359	53.27 ± 0.467	47.79 ± 0.499

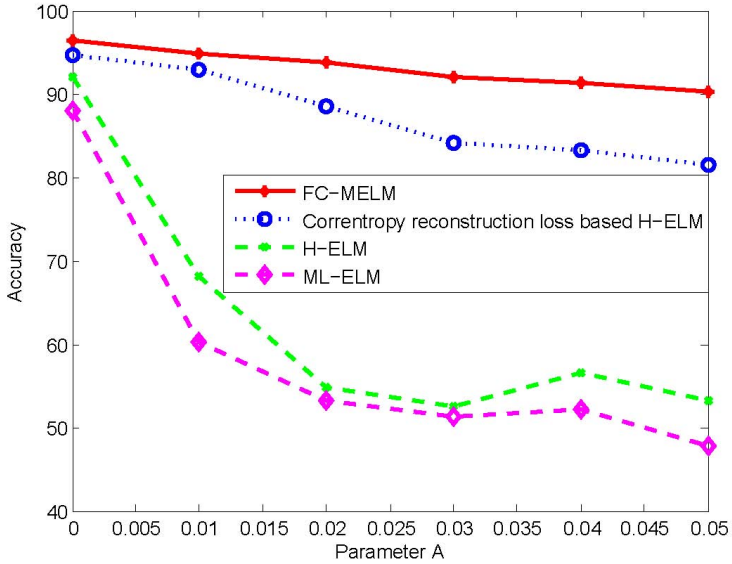

Figure 9: Comparison of classification accuracies of FC-MELM and other methods over WBCD dataset under different levels of impulsive noises.

FC-MELM and H-ELM is presented. Clearly, training speed of FC-MELM is lower than H-ELM. However, FC-MELM still has a short training time over WBCD dataset. We also present the selected bandwidth parameter values of FC-MELM over the WBCD dataset under different noise levels in Table 15.

6. Conclusion and Future Work

In this article, we proposed a novel learning algorithm, called the full correntropy-based multilayer ELM (FC-MELM), which takes advantage of both correntropy and H-ELM. To this end, the conventional MSE-

Table 14: Training Time Comparison of FC-MELM and H-ELM over WBCD dataset

Datasets	WBCD
FC-MELM	0.159s
H-ELM	0.058s

Table 15: Bandwidth parameters of FC-MELM over the WBCD dataset under different noise levels

	$\sigma_{(1)}^{(L1)}$	$\sigma_{(2)}^{(L1)}$	$\sigma_{(1)}^{(L2)}$	$\sigma_{(2)}^{(L2)}$
A=0	0.1	0.009	0.4	0.006
A=0.01	0.6	0.007	0.5	0.008
A=0.02	0.6	0.021	0.7	0.011
A=0.03	0.4	0.010	0.1	0.013
A=0.04	0.5	0.017	0.8	0.015
A=0.05	0.5	0.009	0.4	0.007

based loss function was replaced by the C_{loss}-based loss function in order to improve the robustness. In addition, the conventional ℓ_1-based sparsity penalty term was replaced by a correntropy-based sparsity penalty term. Using correntropy-based sparsity penalty term, a sparser representation of the original data was obtained and further improvement of the robustness was achieved. By conducting simulations on some benchmark datasets under different noises, as well as mislabeled samples, the robustness of the FC-MELM was carefully investigated. Experimental results showed that FC-MELM generally possesses better feature extraction ability than H-ELM and other deep learning methods under corrupted training datasets. These results proved that the proposed algorithm markedly outperformed other algorithms in robustness of outliers and impulsive noises, and also indicated that FC-MELM was much faster than neural network based deep learning algorithms.

The proposed method has some weaknesses that need further investigation, as given in the following. Since the FC-MELM is built in the same spirit as H-ELM, by constructing a multi-layered deep architecture, it inherits some of the fragilities of such architecture. Of particular interest is its performances, which may decay when dealing with increasingly bigger problem spaces and hence bigger parameter spaces. In our extensive experiments, we found that FC-MELM and H-ELM may require important memory ressources, which is not friendly in the era of Big Data analysis. Moreover, although the training speed of FC-MELM is much faster than deep learning algorithms, it remains slower than the original H-ELM. All these weaknesses of the proposed method will be addressed in future work, to reduce the bad influence of increasing parameter

space, to remove the high-memory requirements, and to speed up the training phase. Our goal is to implement the full correntropy-based ELM autoencoder algorithms for real-time applications with lower requirements of hardware platform, such as early cancer diagnosis and on-site medical diagnosis.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61531013), the National Science and Technology Major Project of China (No. 2018ZX03001016) and the Research Fund of Ministry of Education-China Mobile (MCM20150102).

References

[1] Huang, G. B., Zhu, Q. Y., Siew, C. K. Extreme learning machine: theory and applications. Neurocomputing, 70(1) (2006) 489-501.

[2] Zong, W., Huang, G. B. Face recognition based on extreme learning machine. Neurocomputing, 74(16) (2011) 2541-2551.

[3] Mohammed, A. A., Minhas, R., Wu, Q. J., Sid-Ahmed, M. A. Human face recognition based on multidimensional PCA and extreme learning machine. Pattern Recognition, 44(10-11) (2011) 2588-2597.

[4] Cui, D., Huang, G. B. Liu, T. ELM based smile detection using Distance Vector. Pattern Recognition, 79 (2018) 356-369.

[5] Jun, W., Shitong, W., Chung, F. L. Positive and negative fuzzy rule system, extreme learning machine and image classification. International Journal of Machine Learning and Cybernetics, 2(4) (2011) 261-271.

[6] Chacko, B. P., Krishnan, V. V., Raju, G., Anto, P. B. Handwritten character recognition using wavelet energy and extreme learning machine. International Journal of Machine Learning and Cybernetics, 3(2) (2012) 149-161.

[7] Zhang, Y., Wu, J., Zhou, C., Cai, Z. Instance cloned extreme learning machine. Pattern Recognition, 68 (2017) 52-65.

[8] Zheng, W., Qian, Y., Lu, H. Text categorization based on regularization extreme learning machine. Neural Computing and Applications, 22(3-4) (2013) 447-456.

[9] Junior, J. J. D. M. S., Backes, A. R. (2016). ELM based signature for texture classification. Pattern Recognition, 51, 395-401.

[10] Deng, J., Li, K., Irwin, G. W. Fast automatic two-stage nonlinear model identification based on the extreme learning machine. Neurocomputing, 74(16) (2011) 2422-2429.

[11] Wang, H., Qian, G., Feng, X. Q. Predicting consumer sentiments using online sequential extreme learning machine and intuitionistic fuzzy sets. Neural Computing and Applications, 22(3-4) (2013) 479-489.

[12] Huang, G. B., Zhou, H., Ding, X., Zhang, R. Extreme learning machine for regression and multiclass classification. IEEE trans. on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(2) (2012) 513-529.

[13] Frnay, B., Verleysen, M. Parameter-insensitive kernel in extreme learning for non-linear support vector regression. Neurocomputing, 74(16) (2011) 2526-2531.

[14] Xing, H. J., Wang, X. M. Training extreme learning machine via regularized correntropy criterion. Neural Computing and Applications, 23(7-8) (2013) 1977-1986.

[15] Zhao, J., Wang, Z., Park, D. S. Online sequential extreme learning machine with forgetting mechanism. Neurocomputing, 87 (2012) 79-89.

[16] Zhu, Q. Y., Qin, A. K., Suganthan, P. N., and Huang, G. B. Evolutionary extreme learning machine. Pattern recognition, 38(10) (2005) 1759-1763.

[17] Pao, Y. H., Park, G. H., and Sobajic, D. J. Learning and generalization characteristics of the random vector functional-link net. Neurocomputing, 6(2) (1994) 163-180.

[18] Zhang, R., Lan, Y., Huang, G. B., Xu, Z. B. Universal approximation of extreme learning machine with adaptive growth of hidden nodes. IEEE trans. on Neural Networks and Learning Systems, 23(2) (2012) 365-371.

[19] Hinton, G. E., Osindero, S., Teh, Y. W. A fast learning algorithm for deep belief nets. Neural computation, 18(7) (2006) 1527-1554.

[20] Boureau, Y. L., Cun, Y. L. Sparse feature learning for deep belief networks. In Advances in neural information processing systems, 2008, pp. 1185-1192.

[21] Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H. Greedy layer-wise training of deep networks. Advances in neural information processing systems, 2007, pp. 153-160.

[22] Ji, S., Xu, W., Yang, M., Yu, K. 3D convolutional neural networks for human action recognition. IEEE trans. on pattern analysis and machine intelligence, 35(1) (2013) 221-231.

[23] Mobahi, H., Collobert, R., Weston, J. Deep learning from temporal coherence in video. In Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 737-744.

[24] Kasun, L. L. C., Zhou, H., Huang, G. B., Vong, C. M. Representational learning with ELMs for big data. IEEE Intelligent Systems, 28(6) (2013) 31-34.

[25] Tang, J., Deng, C., Huang, G. B. Extreme learning machine for multilayer perceptron. IEEE trans. on neural networks and learning systems, 27(4) (2016) 809-821.

[26] Erdogmus, D., Príncipe, J. C. Information Theoretic Learning, (2009).

[27] Liu, W., Pokharel, P. P., Príncipe, J. C. Correntropy: properties and applications in non-Gaussian signal processing. IEEE trans. on Signal Processing, 55(11) (2007) 5286-5298.

[28] He, R., Zheng, W. S., Hu, B. G., Kong, X. W. A regularized correntropy framework for robust pattern recognition. Neural Computation, 23(8) (2011) 2074-2100.

[29] Wu, Z., Peng, S., Chen, B., Zhao, H. Robust Hammerstein adaptive filtering under maximum correntropy criterion. Entropy, 17(10) (2015) 7149-7166.

[30] Wang, L., Pan, C. Robust level set image segmentation via a local correntropy-based K-means clustering. Pattern Recognition, 47(5) (2014) 1917-1925.

[31] Chen, B., Príncipe, J. C. Maximum correntropy estimation is a smoothed MAP estimation. IEEE Signal Processing Letters, 19(8) (2012) 491-494.

[32] Seth, S., Príncipe, J. C. Compressed signal reconstruction using the correntropy induced metric. In 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 3845-3848.

[33] Singh, A., Príncipe, J. C. A loss function for classification based on a robust similarity metric. In the 2010 International Joint Conference on Neural Networks (IJCNN), 2010, pp. 1-6.

[34] Singh, A., Pokharel, R., Príncipe, J. C. The C-loss function for pattern classification. Pattern Recognition, 47(1) (2014) 441-453.

[35] Qi, Y., Wang, Y., Zheng, X., Wu, Z. Robust feature learning by stacked autoencoder with maximum correntropy criterion. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2014, pp. 6716-6720.

[36] Chen, L., Qu, H., Zhao, J., Chen, B., Príncipe, J. C. Efficient and robust deep learning with correntropy-induced loss function. Neural Computing and Applications, 27(4) (2016) 1019-1031.

[37] Ciresan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. Mitosis detection in breast cancer histology images with deep neural networks. In International Conference on Medical Image Computing and Computer-assisted Intervention, 2013, pp. 411-418.

[38] Litjens, G., Snchez, C. I., Timofeeva, N., Hermsen, M., Nagtegaal, I., Kovacs, I., Van De Kaa, H. C., Bult, P., Van Ginneken, B., and Van Der Laak, J. Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. Scientific reports, 6 (2016) 26286.

[39] Rezaeilouyeh, H., Mollahosseini, A., and Mahoor, M. H. . Microscopic medical image classification framework via deep learning and shearlet transform. Journal of Medical Imaging, 3(4) (2016) 044501.

[40] Kallenberg, M., Petersen, K., Nielsen, M., Ng, A. Y., Diao, P., Igel, C., Vachon, C. M., Holland, K., Winkel, R. R., Karssemeijer, N., and Lillholm, M. Unsupervised deep learning applied to breast density segmentation and mammographic risk scoring. IEEE trans. on medical imaging, 35(5) (2016) 1322-1331.

[41] Xu, Y., Mo, T., Feng, Q., Zhong, P., Lai, M., Eric, I., and Chang, C. . Deep learning of feature representation with multiple instance learning for medical image analysis. In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, 2014, pp. 1626-1630.

[42] Bychkov, D., Turkki, R., Haglund, C., Linder, N., and Lundin, J. Deep learning for tissue microarray image-based outcome prediction in patients with colorectal cancer. In Medical Imaging 2016: Digital Pathology, 2015, pp. 97915-97915.

[43] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural network. Nature, 542 (2017) 115-118.

[44] Schaumberg, A. J., Rubin, M. A., and Fuchs, T. J. H&E-stained Whole Slide Image Deep Learning Predicts SPOP Mutation State in Prostate Cancer. Cold Spring Harbor Laboratory, (2017).

[45] Chang, H., Han, J., Zhong, C., Snijders, A., and Mao, J. H. Unsupervised transfer learning via multi-scale convolutional sparse coding for biomedical applications. IEEE trans. on pattern analysis and machine intelligence, 40(5) (2017) 1182-1194.

[46] Abdel-Zaher, A. M., and Eldeib, A. M. Breast cancer classification using deep belief networks. Expert Systems with Applications, 46 (2016) 139-144.

[47] Fakoor, R., Ladhak, F., Nazi, A., and Huber, M. Using deep learning to enhance cancer diagnosis and classification. In Proceedings of the International Conference on Machine Learning, 28 (2013).

[48] Liang, M., Li, Z., Chen, T., and Zeng, J. . Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. IEEE/ACM trans. on computational biology and bioinformatics, 12(4) (2015) 928-937.

[49] G.-B. Huang, L. Chen, C.-K. Siew. Universal approximation using incremental constructive feed-forward networks with random hidden nodes, IEEE Trans. Neural Networks 17(4) (2006) 879892.

[50] Huang, G. B., Zhou, Hongming, Ding, Xiaojian, and Zhang, Rui. Extreme learning machine for regression and multiclass classification. IEEE trans. on systems, man, and cybernetics. Part B, Cybernetics, 42(2) (2012) 513.

[51] Huang, G. B., Chen, Y. Q., and Babri, H. A. Classification ability of single hidden layer feedforward neural networks. IEEE trans. on Neural Networks, 11(3) (2000) 799-801.

[52] Sandberg, I. W. General structures for classification. IEEE trans. on Circuits and Systems I: Fundamental Theory and Applications, 41(5) (1994) 372-376.

[53] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11) (1998) 2278-2324.

[54] LeCun, Y. , Huang, F. J., Bottou, L. Learning methods for generic object recognition with invariance to pose and lighting. In Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit, 2004, pp. II-97II-104.

[55] Yuan, X. T., Hu, B. G. Robust feature extraction via information theoretic learning. In Proceedings of the 26th annual international conference on machine learning, 2009, pp. 1193-1200.

[56] Beck, A., Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences, 2(1) (2009) 183-202.